Correction - TP 3

Statistiques descriptives univariées

Exercice 1

1. La population Ω étudiée est l'ensemble des matchs de foot durant le tournoi.

   Le caractère X observé est le nombre de but par match.

2. /

3. Voici les commandes :

   - Pour calculer la moyenne : `np.mean(x)` ;
   - Pour calculer la médiane : `np.median(x)` ;
   - Pour calculer la variance : `np.var(x)` ;
   - Pour calculer l'écart type : `np.std(x)` ;
   - Pour calculer l'étendue : `np.max(x) - np.min(x)`.

4. Voici le tableau à compléter :

Modalités	0	1	2	3	4	5	7
Effectifs	3	3	3	6	2	2	1
Fréquences	$\frac{3}{20}$	$\frac{3}{20}$	$\frac{3}{20}$	$\frac{6}{20}$	$\frac{2}{20}$	$\frac{2}{20}$	$\frac{1}{20}$
Fréquences cumulées	$\frac{3}{20}$	$\frac{6}{20}$	$\frac{9}{20}$	$\frac{15}{20}$	$\frac{17}{20}$	$\frac{19}{20}$	$\frac{20}{20}$

5. Le mode vaut 3 (modalité dont l'effectif est le plus grand). Le premier quartile vaut 1, le deuxième 3 et le troisième 3 (qu'on peut déduire avec les fréquences cumulés).

6. On pose `m = np.array([0, 1, 2, 3, 4, 5, 7])` et `n = np.array([3, 3, 3, 6, 2, 2, 1])`.

   On en déduit les fréquences avec la commande `f = n/20` et les fréquences cumulées avec la commande `fcc = np.cumsum(f)`.

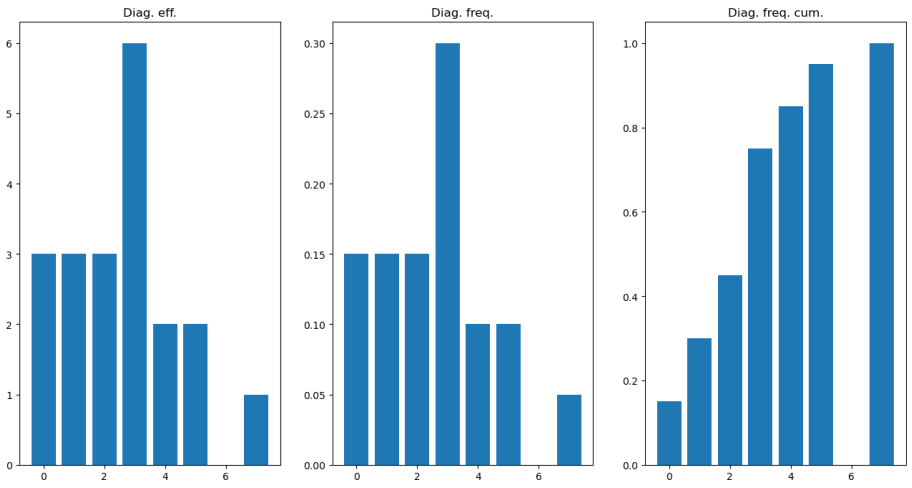
7. `np.sum(n) = 20` car la somme des effectifs est égale à l'effectif total, donc à 20 (il y a 20 matchs).

   `np.sum(f) = 1` car la somme des fréquences est toujours égale à 1.

8. On utilise les commandes suivantes :

```
plt.subplot(1,3,1)
plt.title('Diag. eff.')
plt.bar(m,n)
plt.subplot(1,3,2)
plt.title('Diag. freq.')
plt.bar(m,f)
plt.subplot(1,3,3)
plt.title('Diag. freq. cum.')
plt.bar(m,fcc)
```

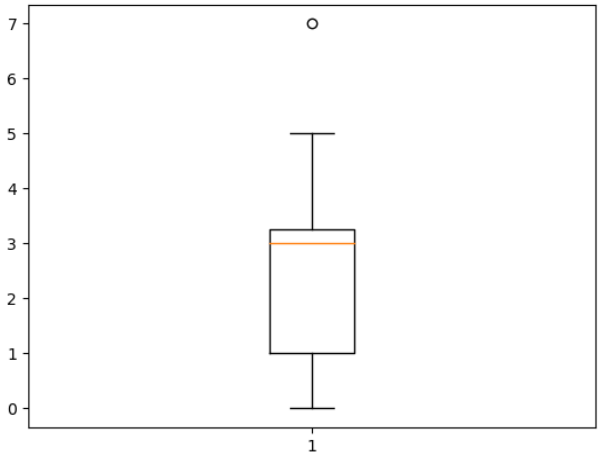
   On obtient le résultat graphique suivant :



9. On utilise les commandes suivantes :

```
plt.boxplot(x)
plt.show()
```

On obtient le résultat graphique suivant :



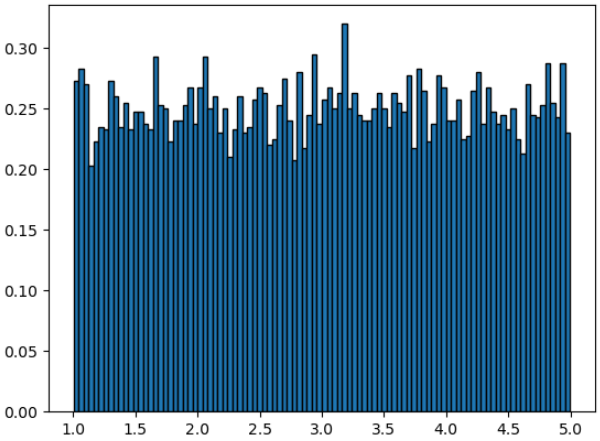
Exercice 2

1. On utilise la commande `4*np.random(10000)+1`.
2. Voici les commandes :
   - Pour calculer la moyenne : `np.mean(x)` ;
   - Pour calculer la médiane : `np.median(x)` ;
   - Pour calculer la variance : `np.var(x)` ;
   - Pour calculer l'écart type : `np.std(x)` ;
   - Pour calculer l'étendue : `np.max(x) - np.min(x)`.
3. Comme la série statistique est constituée de nombres réels aléatoire entre 1 et 5, la probabilité d'obtenir deux fois le même nombre est nulle. Le tri par modalité n'a donc pas d'intérêt ici car on va obtenir 10000 modalités et des effectifs toujours égaux à 1.

   Il faut donc mieux faire un regroupement par classes, en découpant l'intervalle $[1, 5]$ en classes et en rassemblant les modalités par classes.
4. On utilise les commandes suivantes :

```
plt.hist(x,100, density = 'True', edgecolor = 'k')
plt.show()
```

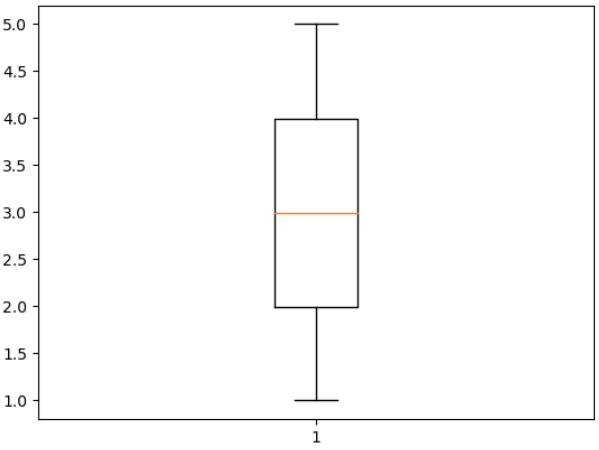
On obtient le résultat graphique suivant :



5. On utilise les commandes suivantes :

```
plt.boxplot(x)
plt.show()
```

On obtient le résultat graphique suivant :



Exercice 3

1. On utilise la commande :

```
>>> df = pd.read_csv("TP3-ex3.csv"); print(df)
```

2. On obtient les indicateurs statistiques correspondant à la population avec la commande :

```
>>> df['Population'].describe()
```

3. On utilise la commande :

```
>>> df['Chomage'].mean()
```

4. On utilise la commande :

```
>>> df.sort_values('Pluie')
```

5. On utilise la commande :

```
>>> df[(df['Densite'] < 4000) & (df['Soleil'] >= 2000)]
```

6. On utilise la commande :

```
>>> df[(df['Region'] == 'PACA') & (df['Fleur'] == 4)]
```

Exercice 4

1. On utilise la commande :

```
>>> df = pd.read_csv("TP3-ex4.csv"); print(df)
```

2. (a) On utilise les commandes :

```
>>> S = df['Superficie'].sum()*1000
>>> P = df['Population'].sum()*1000000
>>> P/S
```

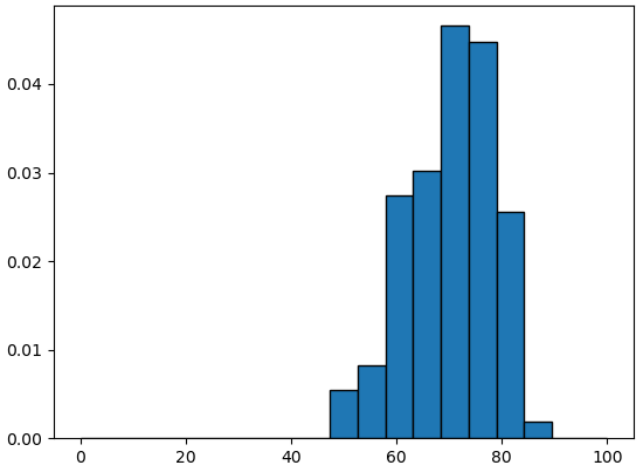
(b) On utilise les commandes :

```
>>> Af = df[df['Index'] <= 57].sum()
>>> AmN = df[(df['Index'] >= 58) & (df['Index'] <= 86)].sum()
>>> AmS = df[(df['Index'] >= 87) & (df['Index'] <= 99)].sum()
>>> Asie = df[(df['Index'] >= 100) & (df['Index'] <= 150)].sum()
>>> Eur = df[(df['Index'] >= 151) & (df['Index'] <= 194)].sum()
>>> Oc = df[df['Index'] >= 195].sum()
>>> S = 1000*np.array([Af[2], AmN[2], AmS[2], Asie[2], Eur[2], Oc[2]])
>>> P = 1000000*np.array([Af[3], AmN[3], AmS[3], Asie[3], Eur[3], Oc[3]])
>>> D = P/S
```

(c) Voici les commandes pour obtenir le diagramme en bâtons demandé :

```
>>> x = np.arange(1,6)
>>> plt.bar(x,D)
>>> plt.show()
```

On obtient le diagramme en bâtons :



3. (a) On utilise les commandes :

```
>>> df['Homme'].mean()
>>> df['Femme'].mean()
```

Ce résultat ne correspond pas cependant à l'espérance de vie moyenne des hommes et des femmes dans le monde : en effet, la Chine et la France par exemple comptent chacune pour 1, alors qu'il faudrait pondérer par le nombre de femmes dans chaque pays. Mais nous n'avons pas accès à ces données malheureusement.

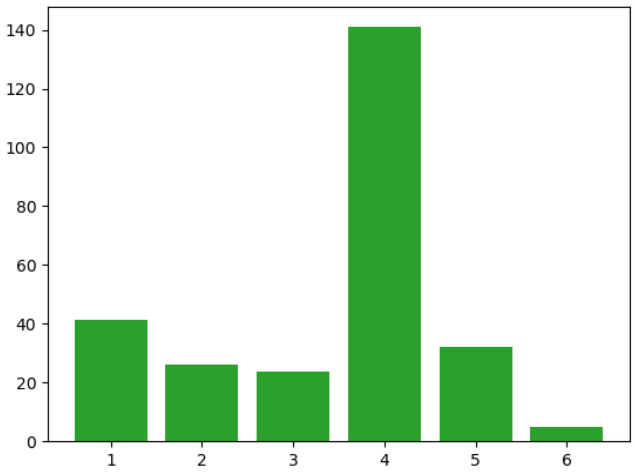
(b) On utilise les commandes (on ferait de même pour les femmes) :

```
>>> df['Homme'].median()
>>> df['Homme'].var()
>>> df['Homme'].std()
```

(c) On utilise les commandes suivantes :

```
>>> h = np.array([df['Homme'][k] for k in range(208)])
>>> c = np.linspace(0,100,20)
>>> plt.hist(h, c, density = 'True', edgecolor = 'k')
>>> plt.show()
```

On obtient l'histogramme :



La classe modale de l'espérance de vie des hommes est donc $[70, 75]$.

(d) On utilise la commande :

```
>>> df.sort_values('Femme')
```

L'espérance de vie des femmes est la plus grande à Monaco (89 ans) et la plus petite en Sierra Léone (51 ans).

(e) On utilise les commandes :

```
>>> df['Femme'].describe()
>>> df[df['Femme'] <= 69]
```

4. (a) On utilise les commandes suivantes :

```
>>> AN = df['Naissance'] - df['Mort']
>>> max(AN)
>>> min(AN)
```

(b) On utilise la commande :

```
>>> df[df['Naissance'] <= df['Mort']]
```

(c) On obtient l'accroissement mondial moyen avec la commande :

```
>>> am = np.mean(AN)
```

On obtient environ $13,1$.

(d) Partons de l'hypothèse d'un taux d'accroissement constant. Notons m le taux d'accroissement de la population mondiale ramené à un habitant, et p la population mondiale. Alors chaque année, la population mondiale change et devient $p = p + m \times p$. On effectue une boucle `for` pour répéter cela jusqu'en 2050 :

```
m = am/1000
p = df['Population'].sum()*1000000
for k in range(2017,2051) :
    p = p+m*p
print(p)
```

On trouve une population de près de 1173 millions d'habitants en 2050. L'INED n'a donc surement pas pris l'hypothèse d'un taux d'accroissement constant.
